

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**FACULTAD DE INGENIERIA CIENCIAS Y SISTEMAS**

**ARQUITECTURA DE COMPUTADORES 1**

**AUX. JORGE GUTIERREZ**

**ING. OTTO LEIVA**



# **Manual Técnico Practica 3 Arduino**

## **INTEGRANTES**

**OSCAR RENE CUELLAR MANCILLA 201503712**

**HAYRTON OMAR IXPATÁ COLOCH 201313875**

**NERY GONZALO GALVEZ GÓMEZ 201403525**

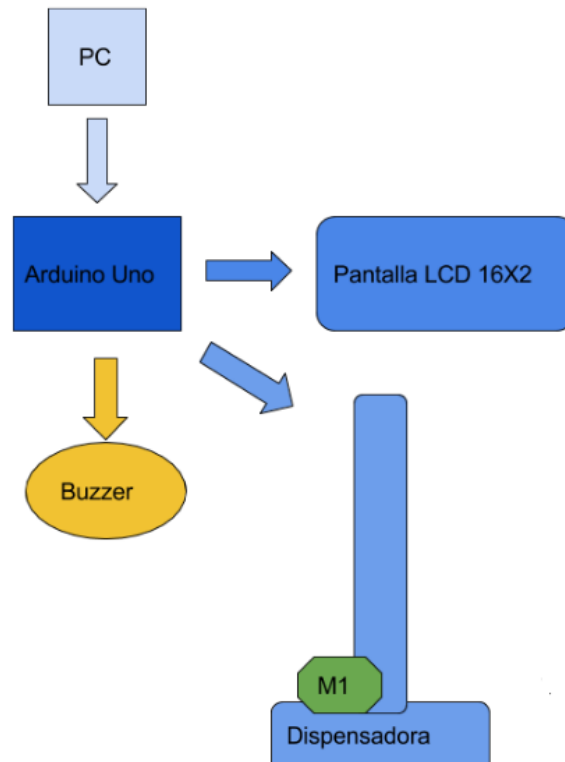
**YOSELIN ANNELICE LEMUS LÓPEZ 201403819**

## **Introducción**

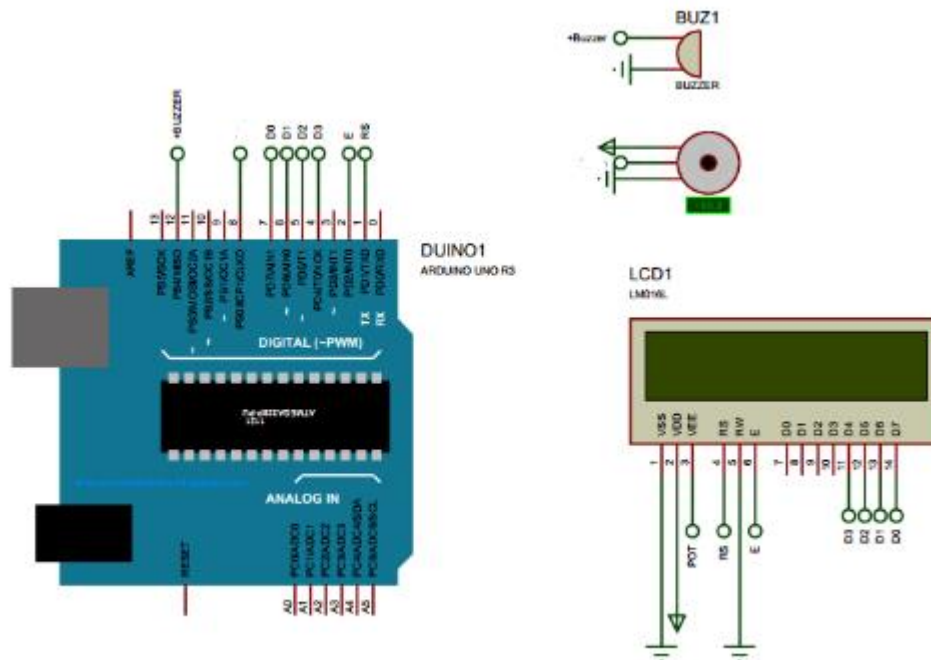
Se presenta la elaboración de una dispensadora de productos, la cual es controlada por una aplicación de escritorio realizada en el lenguaje Java, que además de realizar la comunicación y el envío de solicitudes presenta también una gráfica del tiempo respecto a la cantidad de productos solicitados. A través del proceso de transmisión de datos por conexión serial mediante un puerto USB, la dispensadora despacha los productos cuando detecta una solicitud, estas son almacenadas en una cola, existen dos tipos de solicitud: manual, cuando únicamente se ingresa la cantidad de productos a despachar y automática en la que cada solicitud tiene una hora de despacho, estas últimas, tienen una mayor prioridad sobre las manuales cuando la hora de despacho es la misma que la hora del sistema. La dispensadora fue fabricada con reglas de madera y los productos a dispensar escogido fueron chicles, los cuales serán dispensados mediante un motor que permitirá la salida de los chicles de la dispensadora.

La dispensadora también cuenta con un buzzer que emite alertas sonoras al momento de recibir una solicitud y una alerta diferente al despachar el producto. La dispensadora cuenta también con una pantalla LCD anexa que presentará información como el número de grupo, la cantidad de productos que actualmente tienen la dispensadora, la hora del sistema y la cantidad de solicitudes manuales y automáticas recibidas, todos estos datos son actualizados y presentados automáticamente en la pantalla a medida que se van recibiendo y despachando productos.

## Diagrama de la solución:



## Diagrama de la práctica:



# Explicación de Código:

Para el correcto funcionamiento del programa y de los componentes que se utilizaron en la práctica se utilizaron dos librerías:

Stepper.h: se utilizó para controlar motores stepper, el motor se utilizó para el mecanismo de despacho de la dispensadora.

LiquidCrystal.h: se utilizó para controlar la pantalla LCD donde se mostrara el número de grupo, el tipo de despacho seleccionado, la hora y la cantidad de productos disponibles en la dispensadora.

```
tp3_3$  
//LIBRERIAS  
#include <Stepper.h>  
#include <LiquidCrystal.h>
```

En el setup() se definió el valor de transferencia de datos y la inicialización de la pantalla LCD.

```
void setup() {  
  Serial.begin(9600); //inicializa comunicacion serial  
  lcd.begin(16,2); //inicializa lcd 16x2  
}
```

El método InicializarLCD() se concatenan las cadenas que se mostraran en la dos líneas con que cuenta la pantalla, estas cadenas tendrán el número de grupo el modo en que se encuentra la dispensadora la hora.

```
void InicializarLCD(){  
  if(I_LCD){  
    String l1="!GRUPO 14 A! P ";  
    String l2="00:00 SM SA ";  
    lcd.clear();  
    delay(50);  
    lcd.setCursor(0,0);  
    lcd.print(l1);  
    delay(100);  
    lcd.setCursor(0,1);  
    lcd.print(l2);  
    delay(100);  
    I_LCD=false;  
    Serial.println(l1);  
    Serial.println(l2);  
    delay(2000);  
  }  
}
```

El método Tonos() se ejecutara cuando en el parámetro reciba un carácter que posteriormente al cumplir con uno de los casos reproducirá un todo a través de un buzzer, el tono será distinto para cada función de la dispensadora.

```
//METODO PARA EJECUTAR ALERTAS SONORAS
void Tonos(char s){ //ejecuta los tonos correspondientes a cada solicitud a=automatica , m= manula
    switch(s){
        case 'a':
            for(int i = 0; i<3; i++){
                tone(buz,AUTO[i]);
                delay(100);
            }
            noTone(buz);
            break;
        case 'm':
            for(int i = 0; i<3; i++){
                tone(buz,MANU[i]);
                delay(100);
            }
            noTone(buz);
            break;
        case 'p':
            tone(buz,1700);
            delay(100);
            noTone(buz);
            break;
    }
}
```

El método Dispensar se ejecuta cuando como parámetro recibe un entero el cual es el numero de veces en que se despachara un producto, también se ejecutara el método de Tonos() y se restara uno a las veces en que se despache el producto, los cambios realizados se podrán visualizar en la pantalla LCD.

```
//METODO PARA DISPENSAR PRODUCTOS
void Dispensar(int v){ //ejecuta v veces el metodo de despachar para entregar los n productos solicitud
    for(int i=0; i<v; i++){
        //digitalWrite(led2,1);
        //delay(500);
        Despachar();
        Tonos('p');
        PROD--;
        LCD_P();
        //digitalWrite(led2,0);
        delay(500);
    }
}
```

Los siguientes métodos se utilizan para concatenar los datos que son enviados por la PC y los cambios que se efectuarán al despachar el producto, estos cambios que se realicen podrán ser visualizados en la pantalla LCD.

```
//METODO PARA COLOCAR SERVICIOS MANUALES LCD
void LCD_P(){ //modifica los productos en la LCD
    String pd= "P"+PROD;
    lcd.setCursor(0,13);
    lcd.print(pd);
    delay(100);
}

//METODO PARA COLOCAR HORA LCD
void LCD_H(String h){//Modifica la hora en la LCD
    lcd.setCursor(0,1);
    lcd.print(h);
    delay(100);
}

//METODO PARA COLOCAR SERVICIOS MANUALES LCD
void LCD_PM(){ // modifica las solicitud manual en la LCD
    String pm = "SM"+PM;
    lcd.setCursor(7,1);
    lcd.print(pm);
    delay(100);
}

//METODO PARA COLOCAR SERVICIOS AUTOMATICOS LCD
void LCD_PA(){ //modifica las solicitudes automaticas en la LCD
    String pa = "SA"+PA;
    lcd.setCursor(12,1);
    lcd.print(pa);
    delay(100);
}
```

---

En el Loop() se capturara caracteres que realizaran diferentes funciones, si la el carácter capturado es una 'm' se ejecutara la función manual que posteriormente capturara el cantidad de productos que serán despachados, de la misma forma si el carácter capturado es una 'a', al capturar el carácter 'h' se actualizara la hora en la pantalla LCD esta será enviada desde la PC de forma serial que será capturado por el Arduino.

```

InicializarLCD();
M1.detach();
Tipo = Serial.read();
if(Tipo == 'm'){ //una leta m (char) para poder entrar a modo manual
    //digitalWrite(led,1);
    Serial.println("MANUAL");
    Tonos('m');
    int cantidad = 0;
    while(cantidad == 0 ){ //leera mientras no sea 0 (cantidad de productos)
        cantidad = Serial.parseInt();
        if(cantidad != 0){
            PM++; //suma 1 a la catidad de ordenes entregadas de forma manual
            Serial.println("Producto");
            Dispensar(cantidad);
            LCD_PM();
        }
    }
}else if(Tipo == 'a'){ //una leta a (char) para poder entrar a modo automatico
    //digitalWrite(led3,1);
    Serial.println("AUTOMATICO:");
    Tonos('a');
    int cantidad = 0;
    while(cantidad == 0 ){ //leera mientras no sea 0 (cantidad de productos)
        cantidad = Serial.parseInt();
        if(cantidad != 0){
            PA++; //suma 1 a la catidad de ordenes entregadas de forma automatico
            Serial.println("Producto");
            Dispensar(cantidad);
            LCD_PA();
        }
    }
}

```

---

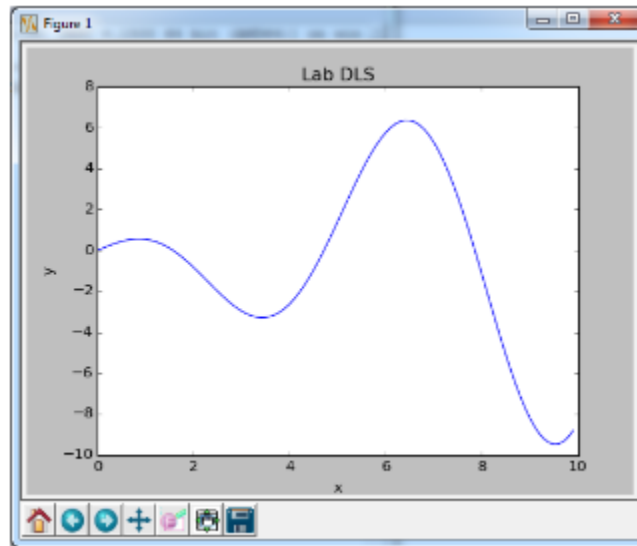
### Gráficas Python:

Para hacer las gráficas de productos vs tiempo necesita tener instaladas las librerías matplotlib y numpy, las cuales pueden encontrarse y descargar de la red sin mayores obstáculos. Debe asegurarse que la versión de las librerías sea compatible con la versión de Python.

```

import numpy as np
x = np.arange(0,10,0.1)
y = x*np.cos(x)
plt.plot(x,y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Lab DLS')
plt.show()

```



## Conclusiones

- La transmisión de datos por conexión serial es muy útil ya que permite enviar información desde una PC a otra que manipula directamente en este caso un microcontrolador que realizará las instrucciones enviadas desde el pc original.
- El lenguaje Python es un buen lenguaje para desarrollar la conexión serial ya que cuenta con librerías accesibles que permiten realizar el proceso de una manera sencilla y potente.
- La implementación de sistemas de control para la automatización de procesos es muy importante, ya que optimiza los recursos con los que se cuentan como el tiempo y los materiales a utilizar, lo que permite un mejor control del sistema desarrollado, con mejores resultados minimizando los errores y costos.